Open Source IaaS OpenStack

Portions of this PPT draw from PPT authored by Professor Dijiang Huang at Arizona State University

Open source IaaS options

- OpenStack
- OpenNebula
- Eucalytpus
- CloudStack

	OpenStack	CloudStack	Eucalyptus	OpenNebula
Source Code	Fully open-source, Apache v2.0	Fully open-source, Apache v2.0	Fully open-source, GPL v3.0	Fully open-source, Apache v2.0

Open source IaaS - OpenStack

- In July 2010 Rackspace Hosting and NASA jointly launched an open-source cloud-software initiative known as OpenStack.
- The community's first official release, code-named Austin, appeared four months later.
- In 2011 developers of the Ubuntu Linux distribution decided to adopt OpenStack.
- In 2012 Red Hat announced a preview of their OpenStack distribution.
- The project is managed by the OpenStack Foundation, a non-profit corporate entity established in September 2012 to promote OpenStack software and its community.
- More than 200 companies joined the project among which are AMD, Brocade Communications Systems, Canonical, Cisco, Dell, EMC, Ericsson, Groupe Bull, HP, IBM, Inktank, Intel, NEC, Rackspace Hosting, Red Hat, SUSE Linux, VMware, and Yahoo!.
- Written in: Python

Open source IaaS - OpenNebula

- OpenNebula is an open-source cloud computing toolkit for managing heterogeneous distributed data center infrastructures.
- The OpenNebula toolkit manages a data center's virtual infrastructure to build private, public and hybrid implementations of infrastructure as a service.
- OpenNebula is sponsored by C12G.
- Developer(s): OpenNebula Community
- Written in: C++, C, Ruby, Java, Shell script, lex, yacc
- Operating system: Linux

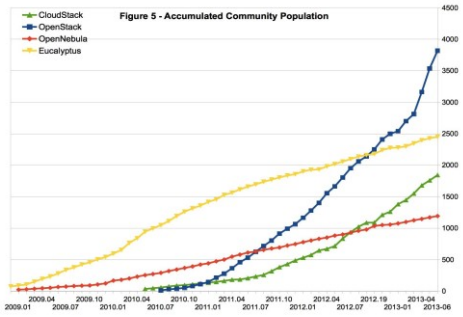
Open source IaaS - Eucalytpus

- Eucalyptus is open source computer software for building Amazon Web Services (AWS)-compatible private and hybrid cloud computing environments marketed by the company Eucalyptus Systems.
- Eucalyptus enables pooling compute, storage, and network resources that can be dynamically scaled up or down as application workloads change.
- Eucalyptus Systems announced a formal agreement with AWS in March 2012 to maintain compatibility.
- Developer(s): Eucalyptus Systems, Inc.
- Written in: Java, C
- Operating system: GNU/Linux, can host Linux and Windows VMs

Open source IaaS - CloudStack

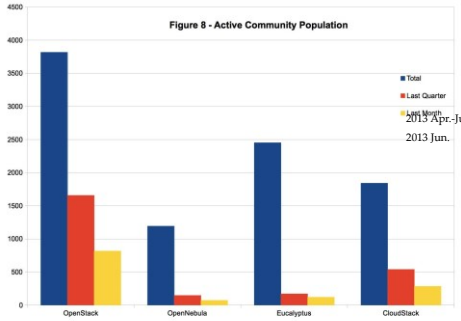
- In May 2010, Cloud.com released most of CloudStack as free software under the GNU General Public License, version 3 (GPLv3).
- Cloud.com and Citrix both supported OpenStack at its announcement in July 2010.
- Citrix purchased Cloud.com on July 12, 2011.
- In August 2011, Citrix released the remaining code under GPLv3.
- In April 2012, Citrix donated CloudStack to the Apache Software Foundation (ASF), where it was accepted into the Apache Incubator; Citrix changed the license to the Apache License version 2. As part of this change, Citrix also ceased their involvement in OpenStack.
- On March 20, 2013, CloudStack graduated from Apache Incubator and became a Top-Level Project (TLP) of ASF.
- Original author(s): Cloud.com, Citrix
- Developer(s): Apache Software Foundation
- Written in: Java (primarily)

Open source IaaS community analysis



7

Open source IaaS community analysis



8

Open source IaaS openness evaluation

The Perspective of the Developer

	OpenStack	CloudStack	Eucalyptus	OpenNebula
Development Model	Public development	Public development	Public development	Public development
Developer Engagement	Contributor license agreement	Contributor license agreement	Contributor license agreement	Contributor license agreement
Governance Model	Foundation	Technical expert	Benevolent dictator	Benevolent dictator

The Perspective of the User

	OpenStack	CloudStack	Eucalyptus	OpenNebula
API Ecosystem	OpenStack API	Amazon API	Amazon API	Amazon API
Production Readiness	No, only available through any of the several vendor specific "stacks"	Enterprise-ready and direct support from developers	Enterprise-ready and direct support from developers	Enterprise-ready and direct support from developers

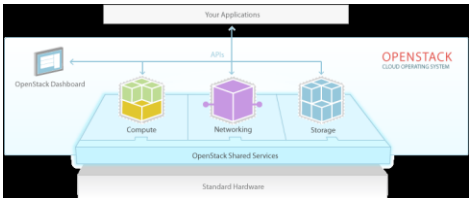
9

OpenStack

Open source IaaS

OpenStack Software

- OpenStack is a cloud operating system that controls large pools of compute, storage, and networking resources throughout a datacenter, all managed through a dashboard that gives administrators control while empowering their users to provision resources through a web interface.



11

OpenStack releases

Series	Status	Releases
Juno	underdevelopment	Due Oct 16, 2014
Icehouse	Current stable release	2014.1
Havana	Security-supported	2013.2
Grizzly	Current stable release, security-supported	2013.1
Folsom	Security-supported	2012.2
Essex	EOL	2012.1
Diablo	EOL	2011.3
Cactus	Deprecated	2011.2
Bexar	Deprecated	2011.1
Austin	Deprecated	2010.1

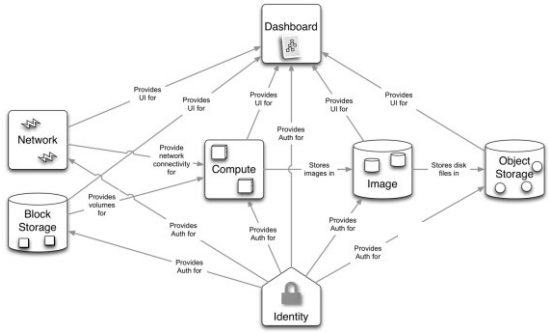
12

OpenStack Components

- There are currently seven core components of OpenStack:
  - Compute
  - Object Storage
  - Identity
  - Dashboard
  - Block Storage
  - Network
  - Image Service

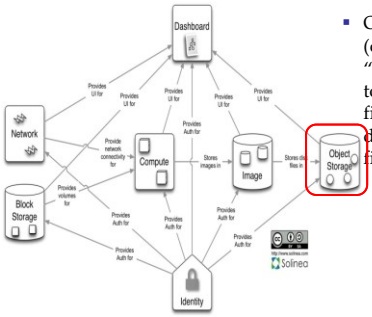
13

Conceptual Architecture



14

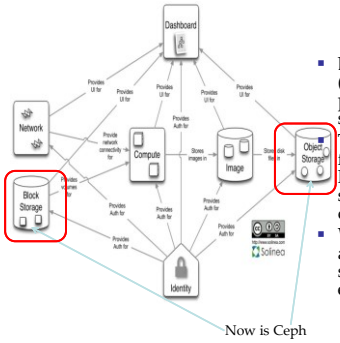
OpenStack Components - Swift



- Object Store (codenamed "Swift") allows you to store or retrieve files (but not mount directories like a fileserver).

15

OpenStack Components - Cinder

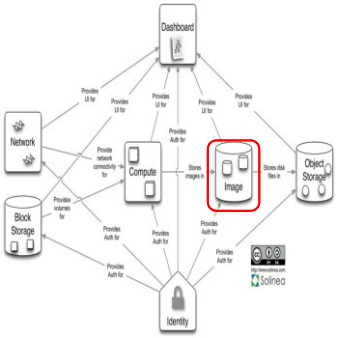


- Block Storage (codenamed "Cinder") provides persistent block storage to guest VMs.
- This project was born from code originally in Nova (the nova-volume service that has been deprecated).
- While this was originally a block storage only service, it has been extended to NFS shares.

Now is Ceph

16

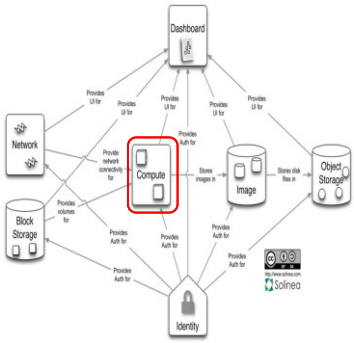
OpenStack Components - Glance



- Image Store (codenamed "Glance") provides a catalog and repository for virtual disk images.
  - These disk images are mostly commonly used in OpenStack Compute.
- Image Store can store the actual virtual disk files in the Object Store.

17

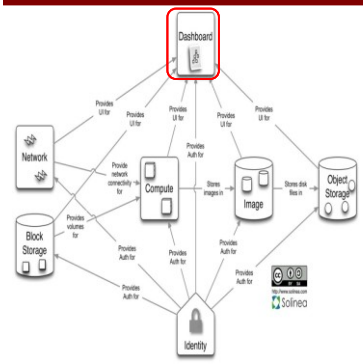
OpenStack Components - Nova



- Compute (codenamed "Nova") provides virtual servers upon demand.
- Compute stores and retrieves virtual disks ("images") and associated metadata in the Image Store ("Glance").

18

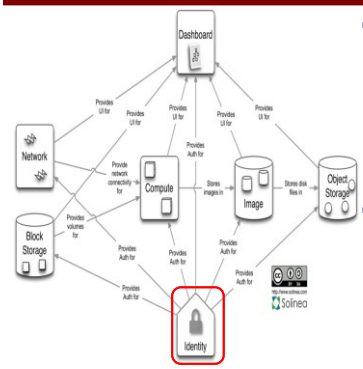
OpenStack Components - Horizon



- Dashboard (codenamed "Horizon") provides a modular web-based user interface for all the OpenStack services.
- With this web GUI, you can perform most operations on your cloud like launching an instance, assigning IP addresses and setting access controls.

19

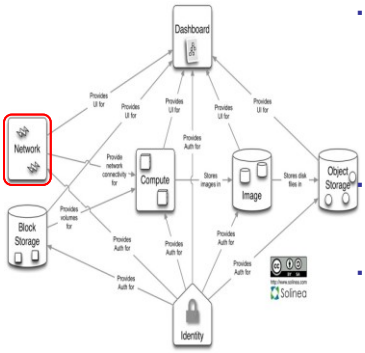
OpenStack Components - Keystone



- Identity (codenamed "Keystone") provides authentication and authorization for all the OpenStack services.
- It also provides a service catalog of services within a particular OpenStack cloud.

20

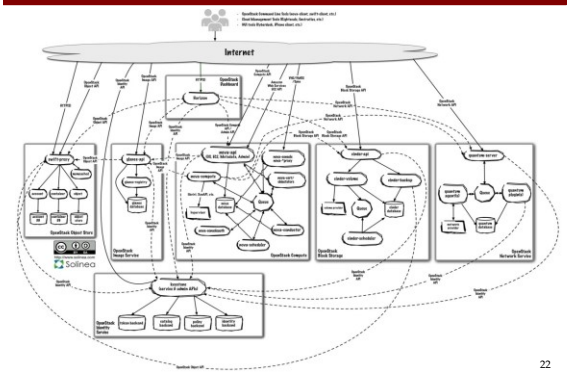
OpenStack Components - Quantum



- Network (which used to named "Neutron" but is in the process of being renamed due to a trademark issue) provides "network connectivity as a service" between interface devices managed by other OpenStack services (most likely Nova).
- The service works by allowing users to create their own networks and then attach interfaces to them.
- Neutron has a pluggable architecture to support many popular networking vendors and technologies.

21

Logical Architecture



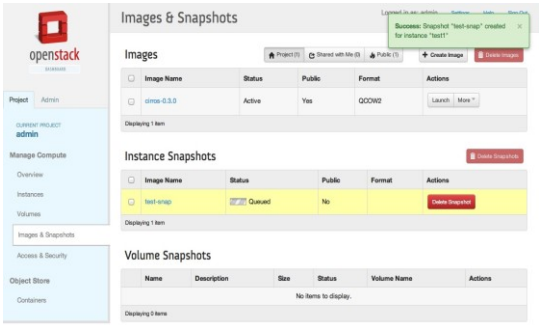
22

Dashboard

- Horizon is a modular Django web application that provides an end user and cloud operator interface to OpenStack services.
- The interface has user screens for:
  - Quota and usage information
  - Instances to operate cloud virtual machines
  - Volume management to control creation, deletion and connectivity to block storage
  - Image and snapshot to upload and control virtual images, which are used to backup and boot new instances
  - Access and security to manage key pairs and security groups (firewall rules)
- In addition to the user screens, it also provides an interface for cloud operators. The operator interface sees across the entire cloud and adds some configuration focused screens such as:
  - Flavors to define service catalog offerings of CPU, memory and boot disk storage
  - Projects to provide logical groups of user accounts
  - Users to administer user accounts
  - System Info to view services running in the cloud and quotas applied to projects

23

Dashboard - demo



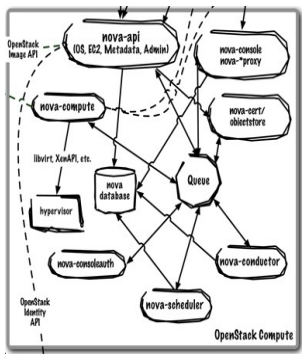
24

Compute

- Nova is the most complicated and distributed component of OpenStack.
- Below is a list of these processes:
  - nova-api
  - nova-compute
  - nova-scheduler
  - nova-conductor
  - nova-console
- Other involved components:
  - queue
  - SQL database

25

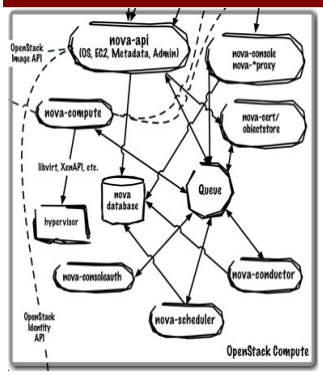
Compute - nova-api



- nova-api is a family of daemons (nova-api, nova-api-os-compute, nova-api-ec2, nova-api-metadata or nova-api-all) that accept and respond to end user compute API calls.
- It supports OpenStack Compute API, Amazon's EC2 API and a special Admin API (for privileged users to perform administrative actions).
- It also initiates most of the orchestration activities (such as running an instance) as well as enforces some policy (mostly quota checks).
- Different daemons allow Nova to implement different APIs (Amazon EC2, OpenStack Compute, Metadata) or combination of APIs (nova-api starts both the EC2 and OpenStack APIs).

26

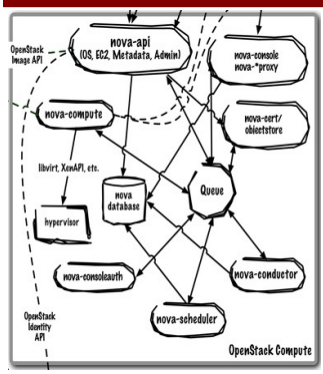
Compute - nova-compute



- The nova-compute process is primarily a worker daemon that creates and terminates virtual machine instances via hypervisor's APIs (XenAPI for XenServer/XCP, libvirt for KVM or QEMU, VMwareAPI for VMware, etc.).
- The process by which it does so is fairly complex but the basics are simple: accept actions from the queue and then perform a series of system commands (like launching a KVM instance) to carry them out while updating state in the database through nova-conductor.

27

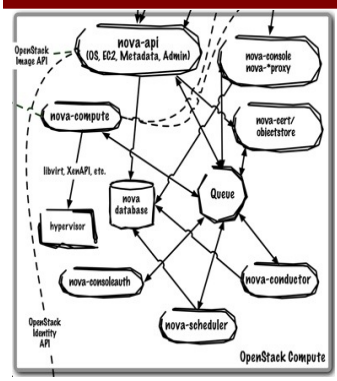
Compute - nova-scheduler



- The nova-scheduler process is conceptually the simplest piece of code in OpenStack Nova: take a virtual machine instance request from the queue and determines where it should run (specifically, which compute server host it should run on).
- In practice, it is now one of the most complex.

28

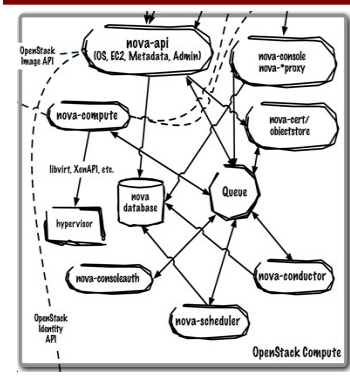
Compute - nova-conductor



- A new service called nova-conductor has been added to this release.
- It mediates access to the database for other daemons (only nova-compute in this release) to provide greater security.

29

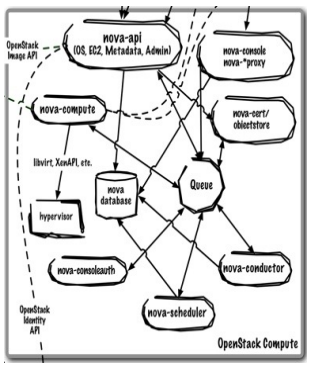
Compute - nova-console



- Nova also provides console services to allow end users to access their virtual instance's console through a proxy.
- This involves several daemons (nova-console, nova-xvpvncproxy, nova-spicehtml5proxy and nova-consoleauth).

30

Compute - queue and database



- The queue provides a central hub for passing messages between daemons. This is usually implemented with RabbitMQ today, but could be any AMPQ message queue (such as Apache Qpid), or Zero MQ.
- The SQL database stores most of the build-time and run-time state for a cloud infrastructure. This includes the instance types that are available for use, instances in use, networks available and projects. Theoretically, OpenStack Nova can support any database supported by SQL-Alchemy but the only databases currently being widely used are sqlite3 (only appropriate for test and development work), MySQL and PostgreSQL.

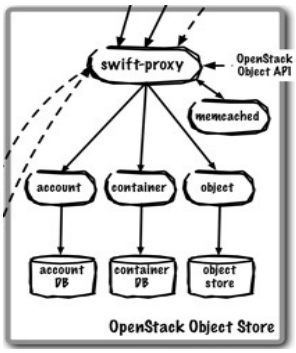
31

Object Store

- OpenStack's Object Store ("Swift") is designed to provide large scale storage of data that is accessible via APIs. Unlike a traditional file server, it is completely distributed, storing multiple copies of each object to achieve greater availability and scalability.
- It includes the following components:
  - Proxy server
  - Account servers
  - Container servers
  - Object servers

32

Object Store



- Proxy server (swift-proxy-server) accepts incoming requests via the OpenStack Object API or just raw HTTP. It accepts files to upload, modifications to metadata or container creation. In addition, it will also serve files or container listing to web browsers. The proxy server may utilize an optional cache (usually deployed with memcache) to improve performance.
- Account servers manage accounts defined with the object storage service.
- Container servers manage a mapping of containers (i.e folders) within the object store service.
- Object servers manage actual objects (i.e. files) on the storage nodes.

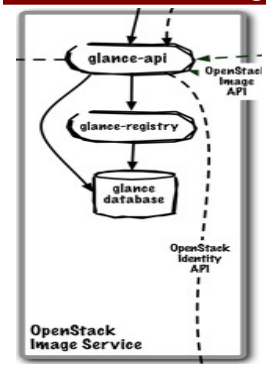
33

Image Store

- OpenStack Image Store centralizes virtual images for users and other cloud services:
  - Stores public and private images that users can utilize to start instances
  - Users can query and list available images for use
  - Delivers images to Nova to start instances
  - Snapshots from running instances can be stored so that virtual machines can be backed

34

Image Store



- glance-api accepts Image API calls for image discovery, image retrieval and image storage.
- glance-registry stores, processes and retrieves metadata about images (size, type, etc.).
- A database to store the image metadata. Like Nova, you can choose your database depending on your preference (but most people use MySQL or SQlite).
- A storage repository for the actual image files. In the diagram above, Swift is shown as the image repository, but this is configurable. In addition to Swift, Glance supports normal filesystems, RADOS block devices, Amazon S3 and HTTP. Be aware that some of these choices are limited to read-only usage.

35

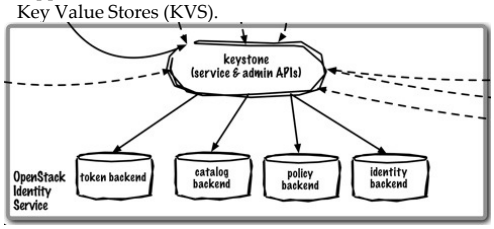
Identity

- Keystone provides a single point of integration for OpenStack policy, catalog, token and authentication:
  - Authenticate users and issue tokens for access to services
  - Store users and tenants for a role-based access control (RBAC)
  - Provides a catalog of the services (and their API endpoints) in the cloud
  - Create policies across users and services

36

Identity

- keystone handles API requests as well as providing configurable catalog, policy, token and identity services.
- Each Keystone function has a pluggable backend which allows different ways to use the particular service. Most support standard backends like LDAP or SQL, as well as Key Value Stores (KVS).



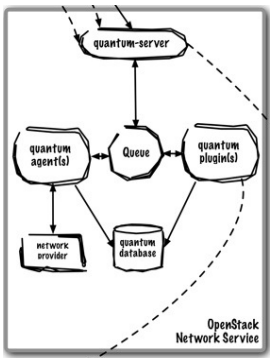
37

Network

- Quantum provides "network connectivity as a service" between interface devices managed by other OpenStack services (most likely Nova).
- Like many of the OpenStack services, Quantum is highly configurable due to it's plug-in architecture. These plug-ins accommodate different networking equipment and software. As such, the architecture and deployment can vary dramatically.

38

Network



- quantum-server accepts API requests and then routes them to the appropriate quantum plugin for action.
- Quantum plugins and agents perform the actual work such as plugging and unplugging ports, creating networks or subnets and IP addressing. These plugins and agents differ depending on the vendor and technologies used in the particular cloud. The common agents are L3 (layer 3), DHCP (dynamic host IP addressing) and vendor specific plug-in agent(s).
- Most Quantum installations will also make use of a messaging queue to route information between the quantum-server and various agents as well as a database to store networking state for particular plugins.

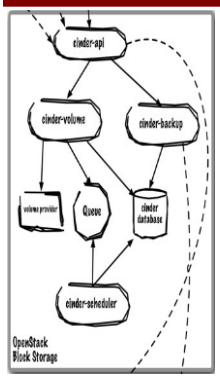
39

Block Storage

- Cinder separates out the persistent block storage functionality that was previously part of OpenStack Compute (in the form of nova-volume) into it's own service. The OpenStack Block Storage API allows for manipulation of volumes, volume types (similar to compute flavors) and volume snapshots:
  - Create, modify and delete volumes
  - Snapshot or backup volumes
  - Query volume status and metadata

40

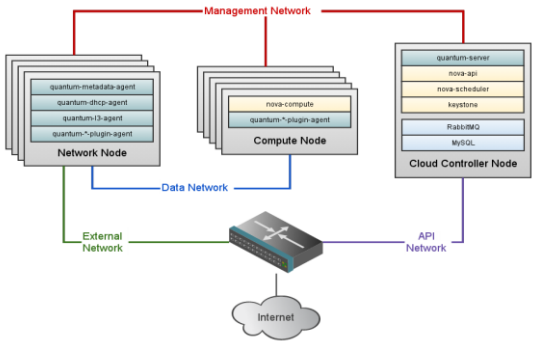
Block Storage



- cinder-api accepts API requests and routes them to cinder-volume for action.
- cinder-volume acts upon the requests by reading or writing to the Cinder database to maintain state, interacting with other processes (like cinder-scheduler) through a message queue and directly upon block storage providing hardware or software. It can interact with a variety of storage providers through a driver architecture.
- Much like nova-scheduler, the cinder-scheduler daemon picks the optimal block storage provider node to create the volume on.
- cinder-backup is a new service that backs up the data from a volume (not a full snapshot) to a backend service. Currently, the only shipping backend service is Swift.
- Cinder deployments will also make use of a messaging queue to route information between the cinder processes as well as a database to store volume state.

41

OpenStack deployment



42

OpenStack deployment

- A standard OpenStack Networking setup has up to four distinct physical data center networks:
  - Management network. Used for internal communication between OpenStack Components. The IP addresses on this network should be reachable only within the data center.
  - Data network. Used for VM data communication within the cloud deployment. The IP addressing requirements of this network depend on the OpenStack Networking plugin in use.
  - External network. Used to provide VMs with Internet access in some deployment scenarios. The IP addresses on this network should be reachable by anyone on the Internet.
  - API network. Exposes all OpenStack APIs, including the OpenStack Networking API, to tenants. The IP addresses on this network should be reachable by anyone on the Internet. This may be the same network as the external network, as it is possible to create a subnet for the external network that uses IP allocation ranges to use only less than the full range of IP addresses in an IP block.

43